Model-free iterative learning control for LTI systems with actuator constraints

Pieter Janssens, Goele Pipeleers and Jan Swevers

Department of Mechanical Engineering, Div. PMA, Katholieke Universiteit Leuven, Celestijnenlaan 300B, Heverlee B3001, Belgium
e-mail: Pieter.Janssens@mech.kuleuven.be.

Abstract: This paper presents a model-free iterative learning control algorithm for linear time-invariant systems. At every trial, a finite impulse response filter to update the system input is calculated by solving a convex optimization problem that minimizes the next trial's tracking error taking into account actuator constraints. Simulation results show the ability of the proposed model-free method to deal with actuator constraints and to fully compensate for trial-invariant disturbances such as actuator cogging.

Keywords: Iterative learning control (ILC), LTI systems, Precision motion control.

1. INTRODUCTION

Iterative learning control (ILC) is an open-loop control strategy that aims at improving the tracking performance of a system executing the same task under the same operating conditions through learning from previous executions (Bristow et al., 2006; Longman, 2000). Using this technique, accurate tracking can be obtained even when the system dynamics are uncertain. ILC has proven to be very efficient in several mechanical applications such as wafer stage motion systems (Heertjes and Tso, 2007) and machine tool axes (Kim and Kim, 1996), which involve linear motor systems that typically suffer from cogging disturbances (Van den Braembussche et al., 1995). Besides motion control, ILC has also been successful in other applications such as active noise control (Stallaert et al., 2010) and chemical process control (Lee et al., 1996).

In ILC design the aim is to use the error information from previous trials as efficiently as possible in order to achieve a minimal tracking error in as few iterations as possible. The simplest ILC law uses a PID-type learning filter, which consists of a proportional, derivative and integral gain on the tracking error. More advanced learning laws (Bristow et al., 2006) include plant inversion methods, quadratically optimal design (Q-ILC), $\mathcal{H}_\infty$ design methods and other optimization-based approaches (Owens and Hätönen, 2005; Nguyen and Banjerdpongchai, 2009; Amann et al., 1996). These methods use a plant model and possibly also uncertainty models to ensure (robust) monotonic convergence.

Contrary to the variety of model-based ILC methods, model-free methods are rare in the ILC literature. Model-free ILC algorithms have the advantage of being applicable to different machines without having to perform identification experiments over and over again. Another shortcoming of existing ILC methods is that only few consider actuator constraints (Mishra et al., 2009). This paper presents an ILC algorithm for single-input single-output (SISO) linear time-invariant (LTI) systems that overcomes both these shortcomings.

The paper is organized as follows. Section 2 details the theory behind the model-free ILC algorithm, analyzes the influence of noise and disturbances on the performance and proposes modifications to the model-free ILC algorithm when noise or disturbances are present. Section 3 presents simulation results of two test cases, which show the effectiveness of the proposed method when dealing with actuator constraints and cogging. Finally, section 4 summarizes the conclusions.

2. A MODEL-FREE ILC ALGORITHM

This section presents the fundamental principle of the model-free ILC algorithm and extends the proposed method for systems with measurement noise and trial-invariant disturbances. At the end of this section the application to closed-loop systems is discussed.

2.1 Fundamental principle of the model-free ILC *algorithm*

Consider the open-loop, SISO, discrete-time, LTI system $P(q)$ in Fig. 1. $P(q)$ has input:

$$u_j(k), \quad k \in \{1, 2, \ldots, N\},$$

output:

$$y_j(k), \quad k \in \{w+1, w+2, \ldots, w+N\},$$

desired output:

$$y_{\mathrm{d}}(k), \quad k \in \{w+1, w+2, \ldots, w+N\},$$

and tracking error $e_j(k) = y_{\mathrm{d}}(k) - y_j(k)$, where subscript $j \in \{0, 1, 2, \ldots\}$ denotes the trial number, k refers to the discrete time instants, q is the one-sample advance operator, w denotes the relative degree of $P(q)$ and N denotes the number of samples per trial.

A widely used ILC update formula is

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k)], \tag{1}$$

where $Q(q)$ is the robustness filter and $L(q)$ the learning filter. Plant information is primarily required for the

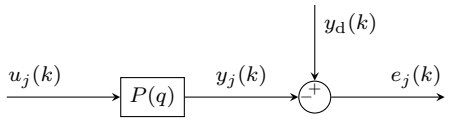

Fig. 1. Open-loop discrete-time LTI system $P(q)$.

design of $L(q)$ (Bristow et al., 2006), which is directly related to the fact that it relates $e_j(k)$ to $u_{j+1}(k)$. This can be understood as follows: suppose that the update law computes $u_{j+1}(k)$ as a linear combination of previous trials' input signals $u_j(k)$, $u_{j-1}(k)$,... only. Then, the corresponding plant output $y_{j+1}(k)$ can be estimated without a plant model, by only relying on its linearity and time-invariance: $y_{j+1}(k)$ will correspond to the same linear combination of $y_j(k)$, $y_{j-1}(k)$,.... Including $e_j(k)$ in the update law (1) compromises this relationship and hereby invokes the need for more plant knowledge. Since we aim for model-free ILC algorithms, we will use update laws of the following general form:

$$u_{j+1}(k) = u_j(k) + u_{\mathrm{lc}}(k) * \alpha_j(k). \tag{2}$$

In this formula $\alpha_j(k)$ denotes the impulse response of a trial-varying, but linear time-invariant, causal finite impulse response filter (FIR-filter) $\alpha_j(q)$ of length N, $u_{\mathrm{lc}}(k)$ represents any linear combination of the previous trials' input signals $u_0(k), u_1(k), \ldots, u_j(k)$, and $*$ denotes the discrete-time convolution operator. When updating the input signal $u_j(k)$ using (2), the corresponding output $y_{j+1}(k)$ is predicted to be:

$$\widehat{y}_{j+1}(k) = y_j(k) + y_{\mathrm{lc}}(k) * \alpha_j(k), \tag{3}$$

where $y_{\mathrm{lc}}(k)$ denotes the corresponding linear combination of previous trials' output signals $y_0(k), y_1(k), \ldots, y_j(k)$, and $\widehat{y}_{j+1}(k)$ denotes the prediction of the output of trial $j+1$ for input $u_{j+1}(k)$, obtained without the use of a system model.

Using the lifted system representation (Phan and Longman, 1988), which is used in the remainder of this paper, the update law (2) is rewritten as:

$$\underbrace{\begin{bmatrix} u_{j+1}(1) \\ u_{j+1}(2) \\ \vdots \\ u_{j+1}(N) \end{bmatrix}}_{\mathbf{u}_{j+1}} = \underbrace{\begin{bmatrix} u_j(1) \\ u_j(2) \\ \vdots \\ u_j(N) \end{bmatrix}}_{\mathbf{u}_j} + \underbrace{\begin{bmatrix} u_{\mathrm{lc}}(1) & 0 & \cdots & 0 \\ u_{\mathrm{lc}}(2) & u_{\mathrm{lc}}(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u_{\mathrm{lc}}(N) & u_{\mathrm{lc}}(N-1) & \cdots & u_{\mathrm{lc}}(1) \end{bmatrix}}_{\mathbf{U}_{\mathrm{lc}}} \underbrace{\begin{bmatrix} \alpha_j(1) \\ \alpha_j(2) \\ \vdots \\ \alpha_j(N) \end{bmatrix}}_{\boldsymbol{\alpha}_j}, \tag{4}$$

where $\mathbf{U}_{\mathrm{lc}}$ denotes the lower-triangular Toeplitz matrix of $u_{\mathrm{lc}}(k)$. Analogous to (4), the predicted output of trial $j+1$ is rewritten as:

$$\widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j + \mathbf{Y}_{\mathrm{lc}}\boldsymbol{\alpha}_j = \mathbf{y}_j + \mathbf{A}_j\mathbf{y}_{\mathrm{lc}}, \tag{5}$$

where

$$\mathbf{Y}_{\mathrm{lc}} = \begin{bmatrix} y_{\mathrm{lc}}(w+1) & 0 & \cdots & 0 \\ y_{\mathrm{lc}}(w+2) & y_{\mathrm{lc}}(w+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ y_{\mathrm{lc}}(w+N) & y_{\mathrm{lc}}(w+N-1) & \cdots & y_{\mathrm{lc}}(w+1) \end{bmatrix}$$

denotes the lower-triangular Toeplitz matrix of $y_{\mathrm{lc}}(k)$ and $\mathbf{A}_j$ denotes the lower-triangular Toeplitz matrix of $\alpha_j(k)$.

Between two trials the trial-varying FIR-filter $\alpha_j(q)$ is computed by solving the following convex optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^N}{\text{minimize}} \quad & \|\mathbf{y}_{\mathrm{d}} - \widehat{\mathbf{y}}_{j+1}\|_2 \\ \text{subject to} \quad & \widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j + \mathbf{Y}_{\mathrm{lc}}\boldsymbol{\alpha}_j \\ & \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_{\mathrm{lc}}\boldsymbol{\alpha}_j \\ & |\mathbf{u}_{j+1}| \leq \overline{u} \\ & |\boldsymbol{\delta u}_{j+1}| \leq \overline{\delta u}. \end{aligned} \tag{6}$$

The ℓ_2-norm of the predicted next trial's tracking error is minimized taking into account linear inequality constraints on $u_{j+1}(k)$ and $\delta u_{j+1}(k) = u_{j+1}(k) - u_{j+1}(k-1)$ to avoid saturation of the actuators. When this convex optimization problem is solved and thus the optimal FIR-filter $\alpha_j(k)$ is known, the next trial's input signal is calculated using (4).

Although any update law of the form (4) allows the output of an LTI system to be predicted without the use of a system model, some particular choices for $u_{\mathrm{lc}}(k)$ result in update laws with important advantages. For now, consider the following simple update laws:

$$u_{\mathrm{lc}}(k) = u_0(k): \quad \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_0\boldsymbol{\alpha}_j, \tag{7}$$

$$u_{\mathrm{lc}}(k) = u_j(k): \quad \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_j\boldsymbol{\alpha}_j. \tag{8}$$

After the first trial, when computing $\mathbf{u}_1$, both update laws are equivalent and make use of the initial input signal $\mathbf{u}_0$. The proposed method, using update law (7) or (8), can be shown to converge in only one iteration to the minimal tracking error, provided that (i) $\mathbf{Y}_0$ is a full-rank matrix, and (ii) no measurement noise or disturbances are present. The first condition is sufficient to ensure that $\mathbf{e}_0 = \mathbf{y}_{\mathrm{d}} - \mathbf{y}_0$ is in the column range of $\mathbf{Y}_0$. The lower-triangular Toeplitz matrix $\mathbf{Y}_0$ is of full rank if and only if $y_0(w+1) \neq 0$, therefore $u_0(1)$ must be nonzero [1]. This way, the condition on the rank of $\mathbf{Y}_0$ restricts the choice of the first trial's input signal $u_0(k)$. The second condition ensures that the predicted output $\widehat{\mathbf{y}}_1 = \mathbf{y}_0 + \mathbf{Y}_0\boldsymbol{\alpha}_0$ is exactly equal to the true output of trial $j = 1$, and the minimal value of the objective function of the optimization problem is the true minimal rms value of the tracking error for the imposed bounds on the actuator input. This globally optimal solution is found since optimization problem (6) is convex.

In practice, however, measurement noise and disturbances are present and more iterations are needed to converge to the optimal system input. At iterations $j > 1$, update law (7) and (8) and hence the corresponding ILC algorithms are different. In the case of update law (7), $\mathbf{Y}_{\mathrm{lc}} = \mathbf{Y}_0$ is of full rank at every trial along the learning process as long as the initial input signal $u_0(k)$ satisfies $u_0(1) \neq 0$. In the case of update law (8), this necessary condition to converge to the optimal system input might not be satisfied further on in the learning process since $\mathbf{Y}_{\mathrm{lc}} = \mathbf{Y}_j$ results from a previous iteration's optimization problem and is not free to choose. For this reason, update law (7) is preferred.

[1] It is assumed that the system's initial conditions are zero.

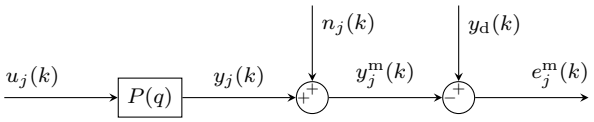

Fig. 2. Open-loop discrete-time system $P(q)$ with measurement noise $n_j(k)$.

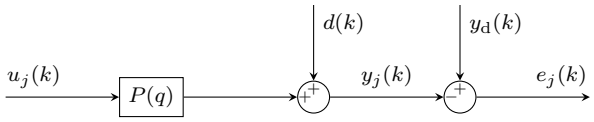

Fig. 3. Open-loop discrete-time system $P(q)$ with a trial-invariant output disturbance $d(k)$.

2.2 Obtaining accurate output predictions in case of measurement noise

This section discusses the influence of measurement noise on the model-free ILC algorithm and proposes an essential modification to the algorithm.

Consider the LTI system $P(q)$ in Fig. 2 with input $u_j(k)$, true output $y_j(k)$, measured tracking error $e_j^{\mathrm{m}}(k)$ and measured output $y_j^{\mathrm{m}}(k) = y_j(k)+n_j(k)$, which is corrupted by zero-mean measurement noise $n_j(k)$ with standard deviation σ_{n}.

Since the true noise-free output $\mathbf{y}_j$ is not known, the predicted plant output $\widehat{\mathbf{y}}_{j+1}$ is computed from the measured output signals $\mathbf{y}_j^{\mathrm{m}}$ and $\mathbf{y}_0^{\mathrm{m}}$:

$$\begin{aligned}\widehat{\mathbf{y}}_{j+1} &= \mathbf{y}_j^{\mathrm{m}} + \mathbf{Y}_0^{\mathrm{m}}\boldsymbol{\alpha}_j,\\ &= \mathbf{y}_j + \mathbf{n}_j + \mathbf{Y}_0\boldsymbol{\alpha}_j + \mathbf{N}_0\boldsymbol{\alpha}_j,\end{aligned} \tag{9}$$

where $\mathbf{N}_0$ and $\mathbf{Y}_0^{\mathrm{m}}$ respectively denote the lower-triangular Toeplitz matrices of $n_0(k)$ and $y_0^{\mathrm{m}}(k)$. The difference between the true output $\mathbf{y}_{j+1}$ and the predicted output $\widehat{\mathbf{y}}_{j+1}$ is called the prediction error of trial $j+1$ and is denoted by $\mathbf{e}_{j+1}^{\mathrm{pr}}$. The update relation (7) still yields

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \mathbf{Y}_0\boldsymbol{\alpha}_j, \tag{10}$$

whereby the prediction error of trial $j+1$ is given by:

$$\mathbf{e}_{j+1}^{\mathrm{pr}} = \mathbf{y}_{j+1} - \widehat{\mathbf{y}}_{j+1} = -\mathbf{n}_j - \mathbf{N}_0\boldsymbol{\alpha}_j. \tag{11}$$

Consequently, the objective function of optimization program (6) amounts to

$$\|\mathbf{y}_{\mathrm{d}} - \widehat{\mathbf{y}}_{j+1}\| = \left\|\mathbf{y}_{\mathrm{d}} - \mathbf{y}_{j+1} + \mathbf{e}_{j+1}^{\mathrm{pr}}\right\|, \tag{12}$$

and hence the prediction error hinders the model-free ILC algorithm from further reducing the tracking error. Therefore it is necessary to constrain this error.

From (11) the standard deviation of the prediction error at sample k of trial $j+1$ can easily be derived:

$$\sigma_{e_{j+1}^{\mathrm{pr}}(k)} = \sigma_{\mathrm{n}}\sqrt{1 + \|\alpha_j(1,\ldots,k)\|_2}. \tag{13}$$

The largest standard deviation of the prediction error is found at the last sample of the trial, $k = N$:

$$\sigma_{e_{j+1}^{\mathrm{pr}}(N)} = \sigma_{\mathrm{n}}\sqrt{1 + \|\boldsymbol{\alpha}_j\|_2}. \tag{14}$$

This analysis shows that adding the convex constraint

$$\|\boldsymbol{\alpha}_j\|_2 \leq t \tag{15}$$

to optimization program (6) limits the standard deviation of the prediction error by $\sigma_{\mathrm{n}}\sqrt{1+t}$ and hence allows the model-free ILC algorithm to further reduce the tracking error.

Hence, in case of measurement noise, the following convex optimization program is solved between two trials to obtain the optimal FIR-filter $\alpha_j(q)$ and thus also the updated input signal $u_{j+1}(k)$:

$$\begin{aligned}\underset{\boldsymbol{\alpha}_j\in\mathbb{R}^N}{\text{minimize}} \quad & \|\mathbf{y}_{\mathrm{d}} - \widehat{\mathbf{y}}_{j+1}\|_2\\ \text{subject to} \quad & \widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j^{\mathrm{m}} + \mathbf{Y}_0^{\mathrm{m}}\boldsymbol{\alpha}_j\\ & \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_0\boldsymbol{\alpha}_j\\ & |\mathbf{u}_{j+1}| \leq \overline{u}\\ & |\boldsymbol{\delta u}_{j+1}| \leq \overline{\delta u}\\ & \|\boldsymbol{\alpha}_j\|_2 \leq t.\end{aligned} \tag{16}$$

In addition to the beneficial effect on the prediction error, the constraint on $\|\boldsymbol{\alpha}_j\|_2$ also limits the change in input signal $\mathbf{U}_0\boldsymbol{\alpha}_j$ from one trial to the next trial. Consequently, the constraint on $\|\boldsymbol{\alpha}_j\|_2$ influences the convergence speed of the ILC algorithm. The upperbound on $\|\boldsymbol{\alpha}_j\|_2$ regulates the trade-off between convergence speed and accuracy of the output prediction.

2.3 Dealing with trial-invariant disturbances

In the previous section the model-free ILC algorithm was adapted for tracking problems where measurement noise is present. In many real-life applications, however, the output also suffers from trial-invariant disturbances. Contrary to measurement noise, these trial-invariant disturbances can be compensated using iterative learning controllers.

Consider again an LTI system $P(q)$ with a certain trial-invariant output disturbance $d_j(k) = d(k)$, for all $j = 0, 1, 2, \ldots$ (see Fig. 3). Using the lifted-system representation, the system dynamics, including the trial-invariant disturbance $d(k)$, are written as follows:

$$\mathbf{y}_j = \mathbf{P}\mathbf{u}_j + \mathbf{d}, \tag{17}$$

where $\mathbf{P}$ denotes the lower-triangular Toeplitz matrix of the impulse response of the system $P(q)$.

Since the model-free ILC algorithm assumes the system dynamics to be LTI, again a prediction error arises in the objective function of the optimization program when update law (7) is used. Combining (17) and (10) results in the predicted output:

$$\begin{aligned}\widehat{\mathbf{y}}_{j+1} &= \mathbf{y}_j + \mathbf{Y}_0\boldsymbol{\alpha}_j,\\ &= \mathbf{P}(\mathbf{u}_j + \mathbf{U}_0\boldsymbol{\alpha}_j) + \mathbf{d} + \mathbf{D}\boldsymbol{\alpha}_j,\end{aligned} \tag{18}$$

where $\mathbf{D}$ denotes the lower-triangular Toeplitz matrix of $d(k)$, whereas the actual output is:

$$\mathbf{y}_{j+1} = \mathbf{P}(\mathbf{u}_j + \mathbf{U}_0\boldsymbol{\alpha}_j) + \mathbf{d}. \tag{19}$$

The resulting prediction error is:

$$\mathbf{e}_{j+1}^{\mathrm{pr}} = \mathbf{y}_{j+1} - \widehat{\mathbf{y}}_{j+1} = \mathbf{D}\boldsymbol{\alpha}_j. \tag{20}$$

Constraining the ℓ_2-norm of $\boldsymbol{\alpha}_j$ would again reduce the prediction error at the cost of convergence speed.

In the presence of trial-invariant disturbances, however, more appropriate choices of the update law, resulting in more accurate output predictions and therefore also faster convergence, can be made.

Consider the following specific case of (4):

$$u_{\mathrm{lc}}(k) = u_j(k) - u_{j-1}(k) + \gamma u_0(k), \tag{21}$$

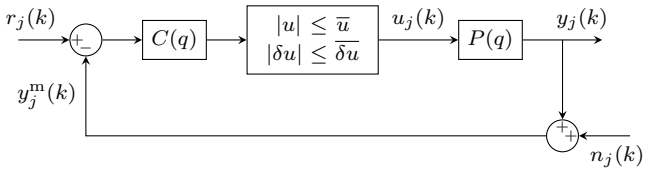

Fig. 4. Closed-loop discrete-time system with actuator constraints.

where $\gamma \approx 0.01 \ldots 0.1$, yielding:

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{A}_j(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0). \tag{22}$$

The predicted output of trial $j+1$ for the system described by (17) is:

$$\begin{aligned} \widehat{\mathbf{y}}_{j+1} &= \mathbf{y}_j + \mathbf{A}_j(\mathbf{y}_j - \mathbf{y}_{j-1} + \gamma \mathbf{y}_0), \\ &= \mathbf{P}\mathbf{u}_j + \mathbf{d} + \mathbf{A}_j(\mathbf{P}(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0) + \gamma \mathbf{d}), \end{aligned} \tag{23}$$

whereas the actual output is:

$$\widehat{\mathbf{y}}_{j+1} = \mathbf{P}(\mathbf{u}_j + \mathbf{A}_j(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0)) + \mathbf{d}. \tag{24}$$

Consequently the resulting prediction error is:

$$\mathbf{e}_{j+1}^{\mathrm{pr}} = \mathbf{y}_{j+1} - \widehat{\mathbf{y}}_{j+1} = \gamma \mathbf{A}_j \mathbf{d}. \tag{25}$$

This analysis shows that update law (22) reduces the prediction error due to trial-invariant disturbances and still allows $\mathbf{Y}_{\mathrm{lc}}$ to be of full rank at every trial along the learning process. The scalar γ is the second tuning parameter of the model-free ILC algorithm next to t, the upperbound on the ℓ_2-norm of $\boldsymbol{\alpha}_j$, and regulates the trade-off between the prediction error due to trial-invariant disturbances and the ability to reduce the next iteration's tracking error.

To summarize, in the presence of measurement noise and trial-invariant disturbances the following convex optimization problem is solved to obtain the optimal FIR-filter and hence also the updated input signal $u_{j+1}(k)$:

$$\begin{aligned} \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^N}{\text{minimize}} \quad & \|\mathbf{y}_{\mathrm{d}} - \widehat{\mathbf{y}}_{j+1}\|_2 \\ \text{subject to} \quad & \widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j^{\mathrm{m}} + (\mathbf{Y}_j^{\mathrm{m}} - \mathbf{Y}_{j-1}^{\mathrm{m}} + \gamma \mathbf{Y}_0^{\mathrm{m}})\boldsymbol{\alpha}_j \\ & \mathbf{u}_{j+1} = \mathbf{u}_j + (\mathbf{U}_j - \mathbf{U}_{j-1} + \gamma \mathbf{U}_0)\boldsymbol{\alpha}_j \\ & |\mathbf{u}_{j+1}| \leq \overline{u} \\ & |\boldsymbol{\delta u}_{j+1}| \leq \overline{\delta u} \\ & \|\boldsymbol{\alpha}_j\|_2 \leq t. \end{aligned} \tag{26}$$

2.4 Application to closed-loop systems with actuator constraints

In most applications, ILC is combined with feedback control since an iterative learning controller cannot compensate for nonrepeating disturbances. Consider the closed-loop system in Fig. 4 with actuator constraints $\overline{u}$ and $\overline{\delta u}$, controller $C(q)$, plant $P(q)$, reference signal $r_j(k)$, actuator input $u_j(k)$, output $y_j(k)$ and measured output $y_j^{\mathrm{m}}(k) = y_j(k) + n_j(k)$. The difference with the aforementioned open-loop systems is that in the closed-loop case the reference signal $r_j(k)$ is updated in order to track a given desired output $y_{\mathrm{d}}(k)$, taking into account the constraints on the actuator input $u_j(k)$, whereas in the open-loop case the actuator input itself is updated.

When the actuator constraints are active and thus the input signal $u_j(k)$ is clipped, the relation between the reference signal $r_j(k)$ and the output $y_j(k)$ of the closed-loop system in Fig. 4 becomes nonlinear. Still, even when the actuator constraints are active, the plant $P(q)$ itself is LTI. Therefore exactly the same optimization program as in the open-loop case can be used to obtain an optimal FIR-filter to construct a predicted next trial's actuator input $\widehat{u}_{j+1}(k)$ that satisfies the actuator constraints. The only required adjustment for closed-loop systems is that the next trial's reference trajectory $r_{j+1}(k)$ is calculated after solving convex optimization problem (26). The next trial's reference signal that corresponds to the predicted next trial's actuator input $\widehat{u}_{j+1}(k)$ and the predicted next trial's output $\widehat{y}_{j+1}(k)$ is given by:

$$\mathbf{r}_{j+1} = \mathbf{C}^{-1}\widehat{\mathbf{u}}_{j+1} + \widehat{\mathbf{y}}_{j+1}, \tag{27}$$

where $\mathbf{C}$ denotes the lower-triangular Toeplitz matrix of the impulse response of controller $C(q)$ and $\widehat{\mathbf{u}}_{j+1}, \widehat{\mathbf{y}}_{j+1}$ are given by an appropriate update law.

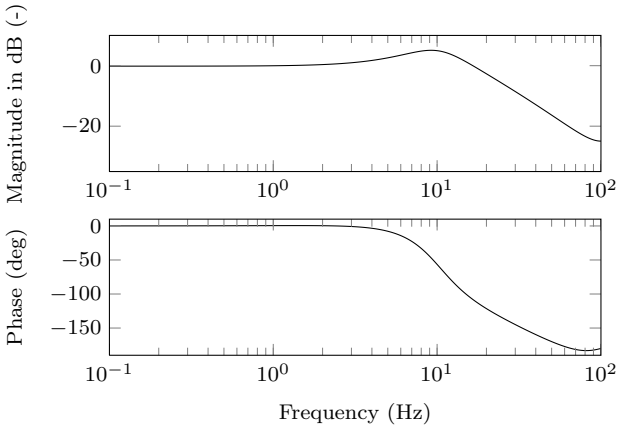

Fig. 5. Bode-diagram of the closed-loop positioning system used for simulation.

3. SIMULATION RESULTS

This section presents the results of two simulation test cases. The first part of this section discusses a test case where a closed-loop system is able to track the desired output without hitting the actuator bounds. This test case shows the ability of the algorithm to fully compensate for cogging disturbances. The second part of this section discusses a test case that shows the ability of the model-free ILC algorithm to deal with actuator constraints. All simulations make use of a discrete-time model of a current controlled linear motor positioning system with an already available position feedback controller. Fig. 5 shows the Bode-diagram of the closed-loop positioning system up to the Nyquist frequency of 100 Hz.

3.1 Cogging compensation

Cogging in permanent-magnet linear motors is an undesirable effect that causes nonsmooth operation of the motor. It is considered as the main disturbance in permanent-magnet linear motors (Ahn, 2005). Cogging is modelled as a combination of two types of ripple: position ripple and force ripple. The position ripple is the necessary force to keep the carriage of the linear motor at a certain position, with zero motor input current. This disturbance force only depends on the position and does not depend on the input current. The force ripple is caused by the variation of the

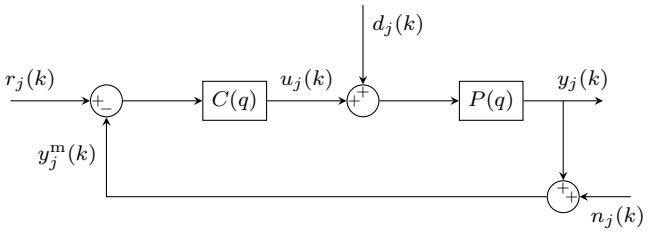

Fig. 6. The closed-loop system with cogging force $d_j(k)$.

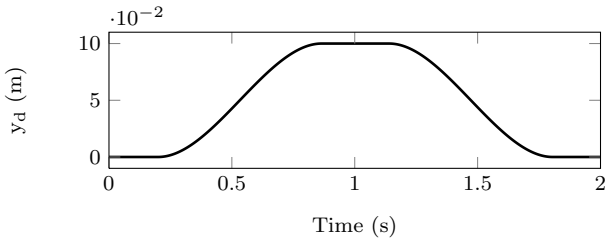

Fig. 7. Desired output $y_{\mathrm{d}}(k)$.

motor constant with the position. Therefore this disturbance force is position-dependent and proportional with the motor input current (Van den Braembussche et al., 1995).

Since the reference trajectory is updated from trial to trial, the cogging disturbance is not entirely trial-invariant. However, the more the algorithm reaches the optimal solution, the smaller the update of the reference trajectory and hence also the trial-to-trial variation of the cogging disturbance will be.

Fig. 6 shows the closed-loop system, with cogging force $d_j(k)$, used in this simulation. The system dynamics, including both force and position ripple, are given by the following equations:

$$\begin{aligned}\mathbf{y}_j = {} & \mathbf{PC}(\mathbf{I}_N + \mathbf{PC})^{-1}(\mathbf{r}_j - \mathbf{n}_j) \\ & + \mathbf{P}(\mathbf{I}_N + \mathbf{PC})^{-1} \Bigg[\underbrace{a\sin(\omega_y \mathbf{y}_j) + b\sin(3\omega_y \mathbf{y}_j + \phi_1)}_{\text{position ripple}} \\ & + \underbrace{\mathbf{u}_j(c\sin(\omega_y \mathbf{y}_j + \phi_2) + d\sin(3\omega_y \mathbf{y}_j + \phi_3))}_{\text{force ripple}} \Bigg], \quad (28a)\end{aligned}$$

$$\mathbf{u}_j = \mathbf{C}(\mathbf{r}_j - \mathbf{y}_j - \mathbf{n}_j), \quad (28b)$$

where $\omega_y = 210\,\mathrm{rad\,m^{-1}}$ denotes the spatial frequency and ϕ_1, ϕ_2, ϕ_3, a, b, c and d are parameters that determine the size and shape of the cogging force.

Fig. 7 shows the desired output $y_{\mathrm{d}}(k)$, which is a forward and backward motion of 10 cm that needs to be executed in 2 seconds. The tracking error to reference signal $y_{\mathrm{d}}(k)$ with and without cogging disturbance is shown in Fig. 8. This figure shows that a significant part of the tracking error is due to the cogging disturbance.

During the first trial of the learning process a reference signal with a rich frequency content, satisfying $r_0(1) \neq 0$, is applied to the system. When calculating the second trial's reference signal, update law (7) is used because experimental data from only one previous iteration are available. From then on, the model-free ILC algorithm uses

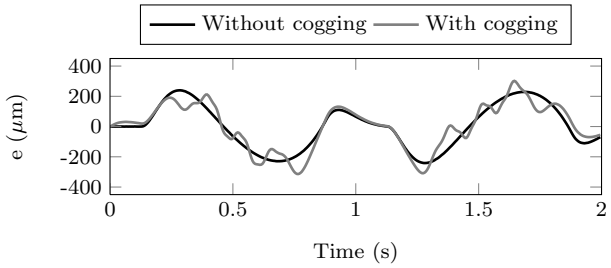

Fig. 8. Tracking error $e(k)$ of the closed-loop system, with and without cogging, when $y_{\mathrm{d}}(k)$ is chosen as reference signal.

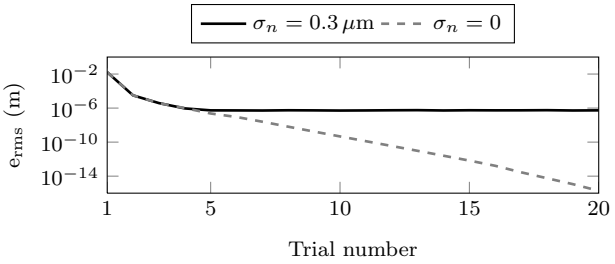

Fig. 9. Rms value of the tracking error as a function of the trial number with and without output noise.

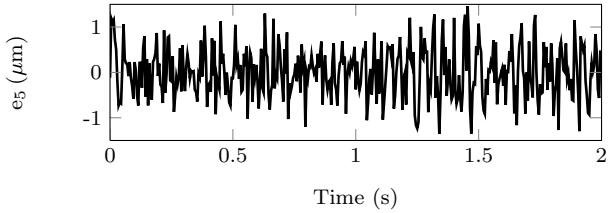

Fig. 10. Tracking error of the 5$^{\mathrm{th}}$ trial in the presence of cogging disturbances and output measurement noise.

the update law given by (22) with $\gamma = 0.05$ and the upperbound t on $\|\alpha_j\|_2$ equal to 1.5.

Two cases are considered: noise-free output measurements ($\sigma_n = 0$) and noise-corrupted output measurements ($\sigma_n = 0.3\,\mu$m). Fig. 9 shows the rms value of the tracking error as a function of the trial number in both these situations. In the presence of measurement noise the rms value of the tracking error converges towards $0.5\,\mu$m in only 5 trials. Fig. 10 shows the tracking error of the 5$^{\mathrm{th}}$ trial. When no noise is present on the output, the model-free ILC algorithm manages to achieve a zero tracking error. These results show that the model-free ILC algorithm is able to compensate for cogging.

3.2 Optimal reference tracking taking into account actuator constraints

The following simulation results illustrate the effectiveness of the proposed method whenever actuator constraints are active. The system dynamics are given by (28a) and

$$\mathbf{u}_j = clip(\mathbf{C}(\mathbf{r}_j - \mathbf{y}_j - \mathbf{n}_j)), \quad (29)$$

where $\mathbf{n}_j$ denotes the measurement noise ($\sigma_n = 0.3\,\mu$m) and $clip(.)$ denotes the nonlinear function, which represents the clipping due to the actuator constraints. The desired output is the same forward and backward motion

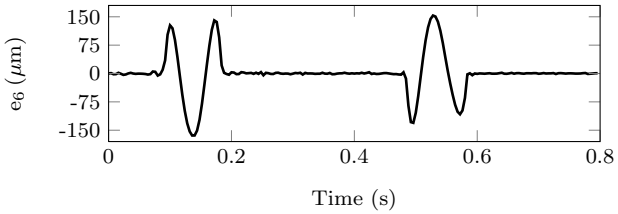

Fig. 11. Tracking error e_6 of the 6^{th} trial.

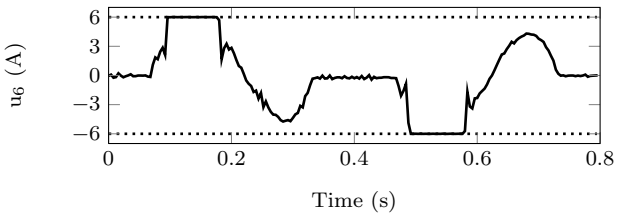

Fig. 12. Actuator input u_6 of the 6^{th} trial together with the bounds on u.

of 10 cm as in the previous example (Fig. 7) but the execution time is reduced to 0.8 seconds such that the constraints on the actuator input become active.

The model-free ILC algorithm, which uses the update law given by (22) with $t = 2$ and $\gamma = 0.05$, reaches convergence in only 6 iterations. Fig. 11 and 12 show respectively the remaining tracking error e_6 and the actuator input u_6 of the 6^{th} iteration. At time instants when no actuator constraints are active the tracking error is of the same order of magnitude as the noise level and cogging is fully compensated. Only when the actuator input hits its bounds, the tracking error is an order of magnitude larger than the noise level.

4. CONCLUSION

This paper presents a model-free ILC algorithm for SISO LTI systems with actuator constraints. The input signal is updated using an optimal trial-varying FIR-filter, which is obtained by solving a convex optimization problem that minimizes the next trial's tracking error. This convex optimization problem allows accounting for linear actuator constraints.

The effect of noise on the next trial's reference signal is reduced by constraining the standard deviation of the prediction error. The upperbound on this convex constraint regulates the trade-off between convergence speed and accuracy of the output prediction. The model-free ILC algorithm is adapted for systems with trial-invariant disturbances using learning laws that yield smaller prediction errors. Simulation results show the ability of the proposed model-free method to deal with cogging disturbances and actuator constraints.

REFERENCES

Ahn, H.S. (2005). State-periodic adaptive compensation of cogging and coulomb friction in permanent-magnet linear motors. *IEEE Transactions on magnetics*, 41, 90–98.

Amann, N., Owens, D., and Rogers, E. (1996). Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proc.-Control Theory Applications*, 143, 217–224.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control: a learning based method for high-performance tracking control. *IEEE Control systems magazine*, 26, 96–114.

Heertjes, M. and Tso, T. (2007). Nonlinear iterative learning control with applications to lithographic machinery. *Control engineering Practice*, 15, 1545–1555.

Kim, D. and Kim, S. (1996). An iterative learning control method with application for cnc machine tools. *IEEE Transactions on Industry Applications*, 32(1), 66–72.

Lee, K., Bang, S., Yi, S., Son, J., and Yoon, S. (1996). Iterative learning control of heat-up phase for a batch polymerization reactor. *Journal of process control*, 4, 255–262.

Longman, R.W. (2000). Iterative learning control and repetitive control for engineering practice. *International journal of control*, 73, 930–954.

Mishra, S., Topcu, U., and Tomizuka, M. (2009). Iterative learning control with saturation constraints. In *American Control Conference*. Hyatt Regency Riverfront, St. Louis, MO, USA.

Nguyen, D.H. and Banjerdpongchai, D. (2009). An LMI approach for robust iterative learning control with quadratic performance criterion. *Journal of process control*, 19, 1054–1060.

Owens, D. and Hätönen, J. (2005). Iterative learning control - an optimization paradigm. *Annual reviews in control*, 29, 57–70.

Phan, M. and Longman, R. (1988). A mathematical theory of learning control for linear discrete multivariable systems. In *Proceedings of the AIAA/AAS Astrodynamics Conference*, 740–746. AIAA Publishers, Minneapolis, Minnesota.

Stallaert, B., Pinte, G., Sas, P., Desmet, W., and Swevers, J. (2010). A novel design strategy for iterative learning and repetitive controllers of systems with a high modal density: Application to active noise control. *Mechanical Systems and Signal Processing*, 24(2), 444–454.

Van den Braembussche, P., Swevers, J., Van Brussel, H., and Vanherck, P. (1995). Motion control of machine tool axes with linear motor. In *Proceedings of the 27th CIRP International Seminar on Manufacturing Systems*. Ann Arbor, Michigan, USA.

ACKNOWLEDGEMENTS

Goele Pipeleers is Postdoctoral Fellow of the Research Foundation - Flanders. This work has been carried out within the framework of projects IWT-SBO 80032 (LECOPRO) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) and G.0422.08 and G.0377.09 of the Research Foundation - Flanders (FWO - Flanders). This work also benefits from K.U.Leuven-BOF EF/05/006 Center-of-Excellence Optimization in Engineering (OPTEC) and from the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office.